

79-80,64

多媒体数据库

连续媒体

12/19

(17)

分布式多媒体数据库中连续媒体的同步

周学海 赵振西

(中国科技大学计算机系 合肥230027)

TP392

摘要 Based on analyzing network delay, we studied smooth representation for the continuous media. In the paper, we present a kind of computing method for the number of media units prefetched and propose a method to compute required buffers in the procedure of playout.

关键词 Continuous media, Low level synchronazition, Buffers, Prefetched units.

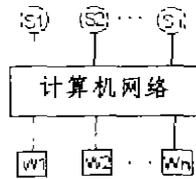
一、引言

在多媒体数据库系统中,多媒体对象的展示显然必不可少,所要展示的媒体可分为三种类型:静态、动态和混合类型。静态类型是指来自不同(或相同)存贮位置,但无时间变化的静态图像、文本等;动态类型是指时间、空间都变化的视频、音频、动画等媒体;混合类型是静态和动态类型的综合,在展示混合和动态类型的媒体时,必须要考虑媒体间以及媒体内部在时间上的同步关系。例如,在同界面上说话者的口形应与其声音保持同步(lip-synchroniza-tion),在同一屏幕的不同窗口显示动画时,应保持各窗口的画面在时间上的同步^[5]。

对于多媒体同步模型,我们可以分为两个层次来研究:高层同步和低层同步。高层同步模型主要研究多媒体时态参数的描述、存贮方法以及在交互式操作中时态参数的更新。它不涉及网络传输延迟、主机处理数据的延迟问题,旨在为低层同步提供展示方案,而在实际的展示算法中,需考虑网络传输、主机处理数据的延迟,为保证同步展示,需要预取一些数据块,要开辟合适的缓冲区及相应的缓冲区管理机制,有时还需对网络传输速率进行控制。本文在文[1]所述的高层同步模型的基础上,对低层同步所涉及的若干问题进行了研究,给出了展示算法中预取块的计算方法,以及缓冲区管理机制。

二、系统模型

对于 n 个媒体服务器 S_1, S_2, \dots, S_n , m 个工作站 W_1, W_2, \dots, W_m , 我们以下图方式连接成一个计算机网络。网上任一工作站 W_i 均可向任一服务器发出请求,不同服务器可通过不同的信道向同一工作站发送信息。



多媒体流数据一般需要分成更小的数据块来传送,以便能更好地控制传输速率,因此需以这些基本数据块为单位,安排多媒体流数据的同步播放。

在实际播放算法中,必须考虑接收方从发出数据申请到收到数据为止的延迟时间,包括:从接收方的较高层向传输层发出请求的时间 d_1 ;从接收方的传输层到发送方的传输层网络传输时间 d_2 ;发送方收到数据请求后获得数据并送到发送方传输的延迟时间 d_3 ;从发送方的传输层转送到接收方的传输层所需时间 d_4 ,对于由 n 块数据块组成的流数据,获得第一块数据所需时间为 $d_1 + d_2 + d_3 + d_4$,获取后续块所需时间为 $d_3 + d_4$ ^[4]。

以上延迟时间,并不是一个确定的常量,而与主机处理的速度、网络负载以及网络传输方式密切相关,因此传输延时很难计算。若要同步播放多媒体流数据,必须采用网络传输延时有界的传输模式,如 FDDI-1 的 MAC 协议,ATM 方式。

我们选定如下一组时间参数来描述多媒体低层同步问题:

$\Delta_{min}^{sr}, \Delta_{max}^{sr}$: 分别表示从接收方申请数据到该申请传送到发送方的最小延时和最大延时,即

$$\Delta_{min}^{sr} = d1_{min} + d2_{min}, \Delta_{max}^{sr} = d1_{max} + d2_{max}.$$

$\Delta_{min}^{rs}, \Delta_{max}^{rs}$: 分别表示从发送方获取数据块并传送到接收方的最小延时和最大延时,即

$$\Delta_{min}^{rs} = d3_{min} + d4_{min}, \Delta_{max}^{rs} = d3_{max} + d4_{max}.$$

$\Delta_{min}^{ss}, \Delta_{max}^{ss}$: 发送两块数据块之间的最小间隔和

最大间隔时间。

- $t[i]$: 第 i 块数据开始发送时刻。
- $a[i]$: 第 i 块数据到达接收方时刻。
- $p[i]$: 第 i 块数据开始播放时刻。
- T_p : 每块数据的播放时间。

三、连续媒体内部的同步播放

播放连续媒体的多媒体对象,其关键要求之一是保证连续媒体流数据的平滑播放,即要使各数据块严格按展示方案播放,如 video 媒体,其录制速率若为 30 帧 (NTSC 制式),那么其播放速率也必须为 30 帧/秒,才能达到原来的录制效果。为达到这一要求,接收方在播放 m 块时,该块数据必须已经传送到接收方,即 $a(\mu) \leq p(\mu)$ 。若假定接收方接收了第一块数据后开始播放,那么为保证媒体的平滑播放,后续块的发送时刻可依据如下定理估算出来。

定理 1 为保证一连续媒体流数据的平滑播放,多媒体流数据发送 μ 块的发送时刻 $t(\mu)$:

$$t(\mu) \leq (\mu - 1)T_p - (\Delta_{max}^m - \Delta_{min}^m) + t(1)$$

证明:为保证 μ 块即时播放,必有 $a^1(\mu) \leq p^0(\mu)$ 。又

$$a^1(\mu) = t(\mu) + \Delta_{max}^m$$

$$p^0(\mu) = p^0(1) + (\mu - 1)T_p \quad (1)$$

而 $p^0(1) = t(1) + \Delta_{max}^m$, 故

$$t(\mu) + \Delta_{max}^m \leq t(1) + \Delta_{max}^m + (\mu - 1)T_p$$

$$t(\mu) \leq (\mu - 1)T_p - (\Delta_{max}^m - \Delta_{min}^m) + t(1) \quad (2)$$

由 (2) 式得:

$$t(\mu) - t(1) \leq (\mu - 1)T_p - (\Delta_{max}^m - \Delta_{min}^m) \quad (3)$$

证毕。 □

根据 (3) 式取 $\mu = 2$ 可得:

$$t(2) - t(1) \leq T_p - (\Delta_{max}^m - \Delta_{min}^m)$$

取 $t(2) - t(1) = \Delta_{min}^m$, 则

$$T_p \geq \Delta_{max}^m + \Delta_{min}^m - \Delta_{min}^m \quad (4)$$

否则,第一块数据播放完后,第二块数据无法到达。当 $T_p < \Delta_{max}^m + \Delta_{min}^m - \Delta_{min}^m$ 时,则需预取若干块数据。

1 预取块的计算

设播放算法预取 N 块数据,即在第 N 块数据到达后开始播放,则根据定理 1 的推导方法可得

$$t(\mu) \leq t(N) + (\mu - 1)T_p - (\Delta_{max}^m - \Delta_{min}^m) \quad (5)$$

$$t(\mu + 1) \leq t(N) + NT_p - (\Delta_{max}^m - \Delta_{min}^m)$$

$$\Delta_{min}^m \leq t(N + 1) - t(N) \leq NT_p - (\Delta_{max}^m - \Delta_{min}^m)$$

$$NT_p \geq \Delta_{max}^m + \Delta_{min}^m - \Delta_{min}^m$$

$$N \geq \frac{\Delta_{max}^m + \Delta_{min}^m - \Delta_{min}^m}{T_p}$$

$$N = \frac{\Delta_{max}^m + \Delta_{min}^m - \Delta_{min}^m}{T_p}$$

以上推导均假设 $\mu > N$ 时,数据块网络传送延时达到最大值 Δ_{max}^m ,数据块发送间隔达到最大值 Δ_{min}^m 时缓冲区的最小值,而实际情况网络延时可能会小于 Δ_{max}^m ,发送间隔也会小于 Δ_{min}^m ,这必然导致接收方所接收的数据越来越多,缓冲区需越来越大。

2 缓冲区的计算方法

为保证媒体流数据的连续播放,需在接收方设置缓冲区,缓冲区至少为预取块的数目,而为消除网络延迟的变化所需缓冲区的大小可由定理 2 计算。

定理 2 为保证 μ 块连续媒体的平滑播放,接收方的缓冲区至多需

$$\frac{(T_p - \Delta_{min}^m)\mu + \Delta_{max}^m - \Delta_{min}^m + (N + 1)\Delta_{min}^m - 1}{T_p} \quad (6)$$

证明:假设 μ' 是 μ 块到达之前最近播放的媒体块,并设预取块数为 N ,则 $p(\mu') \leq a(\mu)$, 又

$$p(\mu') = t(N) + \Delta^m(N) + (\mu' - 1)T_p \leq a(\mu),$$

则

$$\mu' \leq \frac{a(\mu) - (t(N) + \Delta^m(N))}{T_p} + 1$$

$$\mu' \leq \frac{t(\mu) + \Delta_{min}^m(\mu) - (t(N) + \Delta^m(N))}{T_p} + 1$$

要估计缓冲区的最低要求,即要求出在 μ 块最早到达时 μ' 的最小值,因此我们取:

$$\Delta^m(\mu) = \Delta_{min}^m, \Delta^m(N) = \Delta_{max}^m$$

$$\mu'_{min} = \frac{t(\mu) + \Delta_{min}^m - (t(N) + \Delta_{max}^m)}{T_p} + 1$$

$$= \frac{(t(\mu) - t(N)) - (\Delta_{max}^m - \Delta_{min}^m)}{T_p} + 1 \quad (7)$$

又 $(t(\mu) - t(N))_{min} = (\mu - N - 1)\Delta_{min}^m$, 则

$$\mu'_{min} = \frac{(\mu - N - 1)\Delta_{min}^m - \Delta_{max}^m + \Delta_{min}^m}{T_p} + 1$$

$$\mu - \mu' \leq \mu - \frac{(\mu - N - 1)\Delta_{min}^m - \Delta_{max}^m + \Delta_{min}^m}{T_p} - 1$$

$$= \frac{(T_p - \Delta_{min}^m)\mu + \Delta_{max}^m - \Delta_{min}^m + (N + 1)\Delta_{min}^m - 1}{T_p}$$

□

缓冲区的大小 $NB = \mu - \mu'$, 从 (6) 式可知 NB 的大小与 μ 的值成线性关系,即在 $T_p > \Delta_{min}^m$ 时线性增大,这在实际的播放算法中,若 μ 值很大那么 NB 将会很大,这显然是不行的,如何使缓冲区较小,一种可行的方法是控制媒体流数据的发送时刻,发送时刻越迟,要求的缓冲区越小。

因此,在 (7) 式中,我们取 $t(\mu)$ 的最大值,则由 (5) 式知 $t^1(\mu) = t(N) + (\mu - 1)T_p - (\Delta_{max}^m - \Delta_{min}^m)$, 代入 (7) 式得:

$$\mu'_{max} = \mu - \frac{2(\Delta_{max}^m - \Delta_{min}^m)}{T_p} \quad (\text{下转第 64 页})$$